(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 January 2001 (18.01.2001)

PCT

(10) International Publication Number
WO 01/05075 A1

(51) International Patent Classification[7]: H04H 1/00, G11B 20/00

(21) International Application Number: PCT/US00/19481

(22) International Filing Date: 13 July 2000 (13.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/143,432 13 July 1999 (13.07.1999) US

(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052 (US).

(72) Inventors: KIROVSKI, Darko; 16624 NE 34th Court, Redmond, WA 98052 (US). MALVAR, Henrique; 2302 233rd Avenue N.E., Redmond, WA 98053 (US).

(74) Agents: CHRISTIE, Kasey, C. et al.; Suite 500, 421 W. Riverside Avenue, Spokane, WA 99201 (US).

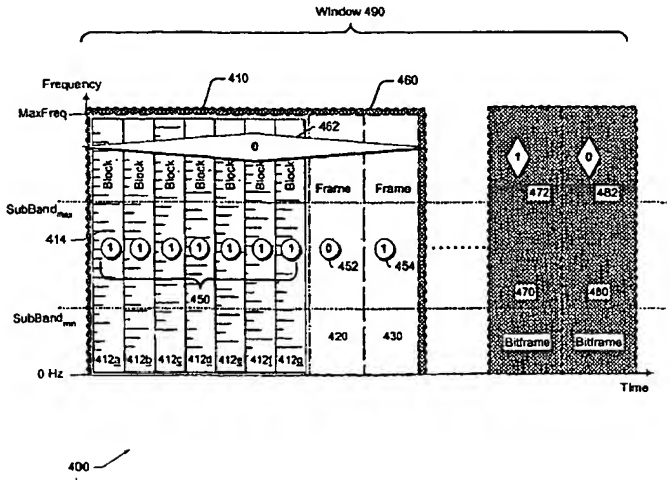
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: IMPROVED AUDIO WATERMARKING WITH COVERT CHANNEL AND PERMUTATIONS



(57) Abstract: Described herein is audio watermarking technology for inserting and detecting watermarks in audio signals, such as a music clip. The watermark identifies the content producer, providing a signature that is embedded in the audio signal and cannot be removed. The watermark is designed to survive all typical kinds of processing and malicious attacks. In one described implementation, a watermarking system employs cover channel encoder to layer an additional information data message on top of the watermark. Thus, an informational message is imposed upon the existing watermark encoded in a signal. In another described implementation, a watermarking system employs a permutation technique to further hide the watermark and it may hide the covert message within the watermark. The order in which data is imposed or encoded is rearranged based upon a permutation table. The same table is used to reverse permute the data at the detector.

WO 01/05075 A1

IMPROVED AUDIO WATERMARKING WITH COVERT CHANNEL AND PERMUTATIONS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from U.S. Provisional Patent Application Serial No. 60/143432 entitled "Improved Audio Watermarking" filed on July 13, 1999.

TECHNICAL FIELD

This invention relates to protecting audio content by using watermarks. More particularly, this invention relates to improved techniques for inserting a covert channel into a signal and for permuting data, such as data of a covert channel inserted into such a signal.

BACKGROUND OF THE INVENTION

Since the earliest days of human civilization, music has existed at the crossroads of creativity and technology. The urge to organize sound has been a constant part of human nature, while the tools to make and capture the resulting music have evolved in parallel with human mastery of science.

Throughout the history of audio recordings, the ability to store and transmit audio (such as music) has quickly evolved since the early days just 130 years ago. From Edison's foil cylinders to contemporary technologies (such as DVD-Audio, MP3, and the Internet), the constant evolution of prerecorded audio delivery has presented both opportunity and challenge.

Music is the world's universal form of communication, touching every person of every culture on the globe. Behind the music is a growing multi-billion

dollar per year industry. This industry, however, is constantly plagued by lost revenues due to music piracy.

Protecting Rights

Piracy is not a new problem. However, as technologies change and improve, there are new challenges to protecting music content from illicit copying and theft. For instance, more producers are beginning to use the Internet to distribute music content. In this form of distribution, the content merely exists as a bit stream which, if left unprotected, can be easily copied and reproduced.

At the end of 1997, the International Federation of the Phonographic Industry (IFPI), the British Phonographic Industry, and the Recording Industry Association of America (RIAA) engaged in a project to survey the extent of unauthorized use of music on the Internet. The initial search indicated that at any one time there could be up to 80,000 infringing MP3 files on the Internet. The actual number of servers on the Internet hosting infringing files was estimated to 2,000 with locations in over 30 countries around the world.

Each day, the wall impeding the reproduction and distribution of infringing digital audio clips (e.g., music files) gets shorter and weaker. "Napster" is an example of an application that is weakening the wall of protection. It gives individuals access to one another's MP3 files by creating a unique file-sharing system via the Internet. Thus, it encourages illegal distribution of copies of copyrighted material.

As a result, these modern digital pirates effectively rob artists and authors of music recordings of their lawful compensation. Unless technology provides for those who create music to be compensated for it, both the creative community and the musical culture at large will be impoverished.

Identifying a Copyrighted Work

Unlike tape cassettes and CDs, a digital music file has no jewel case, label, sticker, or the like on which to place the copyright notification and the identification of the author. A digital music file is a set of binary data without a detectible and unmodifiable label.

Thus, musical artists and authors are unable to inform the public that a work is protected by adhering a copyright notice to the digital music file. Furthermore, such artists and authors are unable to inform the public of any addition information, such as the identity of the copyright holder or terms of a limited license.

Digital Tags

The music industry and trade groups were especially concerned by digital recording because there is no generation loss in digital transfers—a copy sounds the same as the original. Without limits on unauthorized copying, a digital audio recording format could easily encourage the pirating of master-quality recordings.

One solution is to amend an associated digital "tag" with each audio file that identified the copyright holder. To implement such a plan, all devices capable of such digital reproduction must faithfully reproduce the amended, associated tag.

With the passage of the Audio Home Recording Act of 1992, inclusion of serial copying technology became law in the United States. This legislation mandated the inclusion of serial copying technology, such as SCMS (Serial Copy Management System), in consumer digital recorders. SCMS recognizes a "copyright flag" encoded on a prerecorded original (such as a CD), and writes that flag into the subcode of digital copies (such as a transfer from a CD to a DAT tape). The presence of the flag prevents an SCMS-equipped recorder from digitally copying the copy, thus breaking the chain of perfect digital cloning.

However, subsequent developments—both technical and legal—have demonstrated the limited benefits of this legislation. While digital secure music delivery systems (such as SCMS) are designed to support the rights of content owners in the digital domain, the problem of analog copying requires a different approach. In the digital domain, information about the copy status of a given piece of music may be carried in the subcode, which is separate information that travels along with the audio data. In the analog domain, there is no subcode; the only place to put the extra information is to hide it within the audio signal itself.

Digital Watermarks

Techniques for identifying copyright information of digital audio content that address both analog and digital copying instances have received a great deal of attention in both the industrial community and the academic environment. One of the most promising "digital labeling" techniques is augmentation of a digital watermark into the audio signal itself by altering the signal's frequency spectrum such that the perceptual characteristics of the original recording are preserved.

In general, a "digital watermark" is a pattern of bits inserted into a digital image, audio, or video file that identifies the file's copyright information (author, rights, etc.). The name comes from the faintly visible watermarks imprinted on stationery that identify the manufacturer of the stationery. The purpose of digital watermarks is to provide copyright protection for intellectual property that is in digital format.

Unlike printed watermarks, which are intended to be somewhat visible, digital watermarks are designed to be completely invisible, or in the case of audio clips, inaudible. Moreover, the actual bits representing the watermark must be scattered throughout the file in such a way that they cannot be identified and

manipulated. And finally, the digital watermark must be robust enough so that it can withstand normal changes to the file, such as reductions from lossy compression algorithms.

Satisfying all these requirements is no easy feat, but there are several competing technologies. All of them work by making the watermark appear as noise—that is, random data that exists in most digital files anyway. To view a watermark, you need a special program or device (i.e., a "detector") that knows how to extract the watermark data.

Herein, such a digital watermark may be simply called a "watermark." Generically, it may be called an "information pattern of discrete values." The audio signal (or clip) in which a watermark is encoded is effectively "noise" in relation to the watermark.

Watermarking

Watermarking gives content owners a way to self-identify each track of music, thus providing proof of ownership and a way to track public performances of music for purposes of royalty distribution. It may also convey instructions, which can be used by a recording or playback device, to determine whether and how the music may be distributed. Because that data can be read even after the music has been converted from digital to an analog signal, watermarking can be a powerful tool to defeat analog circumvention of copy protection.

The general concept of watermarking has been around for at least 30 years. It was used by companies (such as Muzak™) to *audibly* identify music delivered through their systems. Today, however, the emphasis in watermarking is on *inaudible* approaches. By varying signals embedded in analog audio programs, it is

possible to create patterns that may be recognized by consumer electronics devices or audio circuitry in computers.

For general use in the record industry today, watermarking must be completely inaudible under all conditions. This guarantees the artistic integrity of the music. Moreover, it must be robust enough to survive all forms of attacks. To be effective, watermarks must endure processing, format conversion, and encode/detect cycles that today's music may encounter in a distribution environment that includes radio, the Web, music cassettes, and other non-linear media. In addition, it must endure malevolent attacks by digital pirates.

Watermark Encoding

Typically, existing techniques for encoding a watermark within discrete audio signals facilitate the insensitivity of the human auditory system (HAS) to certain audio phenomena. It has been demonstrated that, in the temporal domain, the HAS is insensitive to small signal level changes and peaks in the pre-echo and the decaying echo spectrum.

The techniques developed to facilitate the first phenomenon are typically not resilient to de-synch attacks. Due to the difficulty of the echo cancellation problem, techniques that employ multiple decaying echoes to place a peak in the signal's cepstrum can hardly be attacked in real-time, but fairly easy using an off-line exhaustive search. (The term "cepstrum" is the accepted terminology for the inverse Fourier transform of the logarithm of the power spectrum of a signal.)

Watermarking techniques that embed secret data in the frequency domain of a signal facilitate the insensitivity of the HAS to small magnitude and phase changes. In both cases, a publisher's secret key is encoded as a pseudo-random sequence that is used to guide the modification of each magnitude or phase

component of the frequency domain. The modifications are performed either directly or shaped according to the signal's envelope.

In addition, watermarking schemes have been developed which facilitate the advantages but also suffers from the disadvantages of hiding data in both the time and frequency domain. It has not been demonstrated whether spread-spectrum watermarking schemes would survive combinations of common attacks: de-synchronization in both the temporal and frequency domain and mosaic-like attacks.

Watermark Detection

The copy detection process is performed by synchronously correlating the suspected audio clip with the watermark of the content publisher. A common pitfall for all watermarking systems that facilitate this type of data hiding is intolerance to desynchronization attacks (e.g., sample cropping, insertion, repetition, variable pitch-scale and time-scale modifications, audio restoration, and arbitrary combinations of these attacks) and deficiency of adequate techniques to address this problem during the detection process.

Desiderata of Watermarking Technology

Watermarking technology has several highly desirable goals (i.e., desiderata) to facilitate protection of copyrights of audio content publishers. Below are listed several of such goals.

Perceptual Invisibility. The embedded information should not induce audible changes in the audio quality of the resulting watermarked signal. The test of perceptual invisibility is often called the "golden ears" test.

Statistical Invisibility. The embedded information should be quantitatively imperceptive for any exhaustive, heuristic, or probabilistic attempt to detect or remove the watermark. The complexity of successfully launching such attacks should be well beyond the computation power of publicly available computer systems.

Tamperproofness. An attempt to remove the watermark should damage the value of the music well above the hearing threshold.

Cost. The system should be inexpensive to license and implement on both programmable and application-specific platforms.

Non-disclosure of the Original. The watermarking and detection protocols should be such that the process of proving audio content copyright both in-situ and in-court, does not involve usage of the original recording.

Enforceability and Flexibility. The watermarking technique should provide strong and undeniable copyright proof. Similarly, it should enable a spectrum of protection levels, which correspond to variable audio presentation and compression standards.

Resilience to Common Attacks. Public availability of powerful digital sound editing tools imposes that the watermarking and detection process is resilient to attacks spawned from such consoles. The standard set of plausible attacks is itemized in the Request for Proposals (RFP) of IFPI (International Federation of the Phonographic Industry) and RIAA (Recording Industry Association of America). The RFP encapsulates the following security requirements:

- two successive D/A and A/D conversions,
- data reduction coding techniques such as MP3,
- adaptive transform coding (ATRAC),
- adaptive subband coding,

- Digital Audio Broadcasting (DAB),
- Dolby AC2 and AC3 systems,
- applying additive or multiplicative noise,
- applying a second Embedded Signal, using the same system, to a single program fragment,
- frequency response distortion corresponding to normal analogue frequency response controls such as bass, mid and treble controls, with maximum variation of 15 dB with respect to the original signal, and
- applying frequency notches with possible frequency hopping.

Watermark Circumvention

If the encoding of a watermark can thwart a malicious attack, then it can avoid the harm of the introduction of unintentional noise. Therefore, any advancement in watermark technology that makes it more difficult for a malevolent attacker to assail the watermark also makes it more difficult for a watermark to be altered unintentionally.

In general, there are two common classes of malevolent attacks:

1. De-synchronization of watermark in digital audio signals. These attacks alter audio signals in such a way to make it difficult for the detector to identify the location of the encoded watermark codes.
2. Removing or altering the watermark. The attacker discovers the location of the watermark and intentionally alters the audio clip to remove or deteriorate a part of the watermark or its entirety.

Framework to Thwart Attacks

Accordingly, there is a need for a new framework of protocols for hiding and detecting watermarks in digital audio signals that are effective against malevolent attacks. The framework should possess several attributes that further the desiderata of watermark technology, described above. For example, such desiderata include "perceptual invisibility" and "statistical invisibility". The framework should be tamperproof and inexpensive to license and implement on both programmable and application-specific platforms. The framework should be such that the process of proving audio content copyrights both in-situ and in-court does not involve usage of the original recording.

The framework should also be flexible to enable a spectrum of protection levels, which correspond to variable audio presentation and compression standards, and yet resilient to common attacks spawned by powerful digital sound editing tools.

SUMMARY OF THE INVENTION

Described herein is an audio watermarking technology for inserting and detecting watermarks in audio signals, such as a music clip. The watermark identifies the content producer, providing a signature that is embedded in the audio signal and cannot be removed. The watermark is designed to survive all typical kinds of processing, including compression, equalization, D/A and A/D conversion, recording on analog tape, and so forth. It is also designed to survive malicious attacks that attempt to remove or modify the watermark from the signal, including changes in time and frequency scales, pitch shifting, and cut/paste editing.

In one described implementation, a watermarking system employs covert channel encoder to layer an additional information data message on top of the

watermark. Thus, an informational message is imposed upon the existing watermark encoded in a signal.

In another described implementation, a watermarking system employs a permutation technique to further hide the watermark and it may hide the covert message within the watermark. The order in which data is imposed or encoded is rearranged based upon a permutation table. The same table is used to reverse permute the data at the detector.

BRIEF DESCRIPTION OF THE DRAWINGS

The same numbers are used throughout the drawings to reference like elements and features.

Fig. 1 is a block diagram of an audio production and distribution system in which a content producer/provider watermarks audio signals and subsequently distributes that watermarked audio stream to a client over a network.

Fig. 2 is a block diagram of a watermarking encoding system implemented, for example, at the content producer/provider.

Fig. 3 is a block diagram of a watermarking detecting unit implemented, for example, at the client.

Figs. 4A-4E show graphs of an audio clip to illustrate blocking, framing, bitframing, and windowing of such audio clip.

Fig. 5 illustrates sample blocks, frames, bitframes, and windows of an audio clip and it further illustrates the encoding of bit values of a watermark within such blocks and frames. It still further illustrates the encoding of covert bit values of a covert message over bitframes.

Fig. 6 illustrates sample bitframes and a window of an audio clip to show a permutation technique of an implementation of watermarking.

Fig. 7 is a flow diagram showing a methodological implementation of watermark encoding.

Fig. 8 is a flow diagram showing a methodological implementation of watermark decoding.

Fig. 9 is an example of a computing operating environment capable of implementing the improved audio watermarking with covert channel and permutations.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The following description sets forth specific embodiments of the improved audio watermarking with covert channel and permutations that incorporate elements recited in the appended claims. These embodiments are described with specificity in order to meet statutory written description, enablement, and best-mode requirements. However, the description itself is not intended to limit the scope of this patent. Rather, the inventors have contemplated that the claimed improved audio watermarking with covert channel and permutations might also be embodied in other ways, in conjunction with other present or future technologies.

Incorporation by Reference

The following provisional application (from which priority is claimed) is incorporated by reference herein: U.S. Provisional Patent Application Serial No. 60/143432 entitled "Improved Audio Watermarking" filed on July 13, 1999.

In addition, the following co-pending patent applications are incorporated by reference herein:

- U.S. Patent Application Serial No. 09/316,899, entitled "Audio Watermarking with Dual Watermarks" filed on May 22, 1999, and assigned to the Microsoft Corporation; and
- U.S. Patent Application Serial No. 09/259,669, entitled "A System and Method for Producing Modulated Complex Lapped Transforms" filed on February 26, 1999, and assigned to the Microsoft Corporation.

The following U.S. Patent is incorporated by reference herein: U.S. Patent No. 6,029,126, entitled "Scalable Audio Coder and Decoder" issued on February 22, 2000, and assigned to the Microsoft Corporation.

Introduction

Described herein are at least two exemplary implementations of improved audio watermarking with covert channel and permutations (i.e., "exemplary watermarking"). The first exemplary watermarking implementation employs covert channel encoder to layer an additional information data message on top of the watermark. Thus, an informational message is imposed upon the existing watermark encoded in a signal. The second exemplary watermarking implementation employs a permutation technique to further hide the watermark and it may hide the covert message within the watermark. The order in which data is imposed or encoded is rearranged based upon a permutation table. The same table is used to reverse permute the data at the detector.

The exemplary watermarking implementations, described herein, may be at least implemented by an audio production and distribution system like that shown in Fig. 1 and by a computing environment like that shown in Fig. 9.

A watermark may be generically called an "information pattern of multiple discrete values" because it is a pattern of binary bits designed to convey information. It may also be referred to as a "data pattern." A watermark is encoded in a digital audio signal (or clip). In relation to the watermark, the audio signal is effectively "noise." In general, watermarking involves hiding the information of the watermark within the "noise" of a digital signal.

Audio Production and Distribution System Employing Watermarks

Fig. 1 shows an audio production and distribution system 20 having a content producer/provider 22 that produces original musical content and distributes the musical content over a network 24 to a client 26. The content producer/provider 22 has a content storage 30 to store digital audio streams of original musical content. The content producer 22 has a watermark encoding system 32 to sign the audio data stream with a watermark that uniquely identifies the content as original. The watermark encoding system 32 may be implemented as a standalone process or incorporated into other applications or an operating system.

A watermark is an array of bits generated using a cryptographically secure pseudo-random bit generator and a new error correction encoder. The pseudo-uniqueness of each watermark is provided by initiating the bit generator with a key unique to each audio content publisher. The watermark is embedded into a digital audio signal by altering its frequency magnitudes such that the perceptual audio characteristics of the original recording are preserved. Each magnitude in the frequency spectrum is altered according to the appropriate bit in the watermark.

The watermark encoding system 32 applies the watermark to an audio signal from the content storage 30. Typically, the watermark identifies the content producer 22, providing a signature that is embedded in the audio signal and cannot

be removed. The watermark is designed to survive all typical kinds of processing, including compression, equalization, D/A and A/D conversion, recording on analog tape, and so forth. It is also designed to survive malicious attacks that attempt to remove the watermark from the signal, including changes in time and frequency scales, pitch shifting, and cut/paste editing.

The content producer/provider 22 has a distribution server 34 that streams the watermarked audio content over the network 24 (e.g., the Internet). An audio stream with a watermark embedded therein represents to a recipient that the stream is being distributed in accordance with the copyright authority of the content producer/provider 22. The server 34 may further compress and/or encrypt the content conventional compression and encryption techniques prior to distributing the content over the network 24.

The client 26 is equipped with a processor 40, a memory 42, and one or more media output devices 44. The processor 40 runs various tools to process the audio stream, such as tools to decompress the stream, decrypt the date, filter the content, and/or apply audio controls (tone, volume, etc.). The memory 42 stores an operating system 50 (such as a Microsoft® Windows 2000® operating system), which executes on the processor. The client 26 may be embodied in a many different ways, including a computer, a handheld entertainment device, a set-top box, a television, an audio appliance, and so forth.

The operating system 50 implements a client-side watermark detecting system 52 to detect watermarks in the audio stream and a media audio player 54 to facilitate play of the audio content through the media output device(s) 44 (e.g., sound card, speakers, etc.). If the watermark is present, the client can identify its copyright and other associated information.

The operating system 50 and/or processor 40 may be configured to enforce certain rules imposed by the content producer/provider (or copyright owner). For instance, the operating system and/or processor may be configured to reject fake or copied content that does not possess a valid watermark. In another example, the system could play unverified content with a reduced level of fidelity.

Watermark Insertion and Detection

In general, Fig. 2 shows a watermark encoding system 100 (or simply "watermark encoder") that may be implemented at a content provider/producer to encode the audio signal with a watermark. The watermark encoding system 100 has a converter 110 to convert an audio signal into frequency and phase components and a mask frequency-processor domain magnitude and phase components. It may also include an energy-level trigger 112 to determine a hearing threshold for corresponding frequency components. The trigger 112 determines whether the energy level across a portion of the signal warrants encoding of the watermark in that portion.

The watermark encoding system 100 also has a pattern generator 114 to generate the watermark, and a watermark insertion unit (WIU) 116 to insert the watermark into the audio signal. It also has a deconverter 118 to convert the audio signal back into the time domain.

Within the WIU 116 (or closely associated therewith) are a pseudorandom number generator (PRNG) 120, a covert message subunit (CMSU) 122, and a permutation subunit (PSU) 124. The PRNG 120 calculates values in a pseudorandom fashion based upon a key. Typically, this key is the same key used to generate the watermark by the pattern generator 114. The CMSU 122 imposes a

covert message onto a watermark. The PSU 124 permutes (i.e., reorders) values encoded into the audio signal.

In general, Fig. 3 shows a watermark detecting system 130 that may be implemented at a client that plays the audio clip (containing the audio signal). Like the encoding system 100, the watermark detecting system 130 has a converter 14012, a mask processor 142, and a watermark pattern generator 144. It is also equipped with a watermark detector 146 that locates a watermark in the audio clip. The watermark detector 146 determines which block interval of the watermarked audio signal contains the watermark pattern and if the watermark generated by a particular key is present in that block interval of the signal.

The watermark encoding system 100 has a deconverter 118 to convert the audio signal back into the time domain. Pseudorandom number generator (PRNG) 120 is employed to implement the exemplary watermarking, but its role is explained later.

In general, Fig. 3 shows a watermark detecting system 130 (or simply "watermark detector") that may be implemented at a client that plays the audio clip (containing the audio signal). Also, it may be implemented in an audio management and distribution subsystem (for example, in an application that downloads music clips from the Internet and uploads them to portable devices).

The watermark detecting system 130 has a converter 140, a mask processor 142, and a watermark pattern generator 144. The converter 140 converts an audio signal that is suspected to include a watermark. It converts the signal into its frequency-domain magnitudes. The mask processor 142 determines the hearing threshold for frequency-domain magnitude components. The pattern generator 144 generates a comparison watermark based upon the same watermark key as used by the encoder. The pattern generator 144 typically includes a pseudorandom number

generator (PRNG) to generate the comparison watermark based upon a watermark key.

The watermark detecting system 130 is also equipped with a watermark detector 146 that locates a watermark in the audio clip. The watermark detector 146 determines which block interval of the watermarked audio signal contains a watermark pattern and whether that discovered watermark pattern matches the comparison watermark generated by the pattern generator 144.

The watermark detection unit 146 typically includes a pseudorandom number generator (PRNG) 150, a covert message subunit (CMSU) 152, and a permutation subunit (PSU) 154. The PRNG 150 calculates values in a pseudorandom fashion based upon a key. That key to generate typically is the same key used to generate the watermark by the pattern generator 114 of the encoder and the pattern generator 144 of the detector. The CMSU 152 extracts the covert message from a watermark. The PSU 154 reverse-permutes (i.e., returns to original order) values encoded into the audio signal.

Blocks, Frames, Bitframes, and Windows

Blocks. During the encoding, the original audio signal is processed into equally sized, overlapping, time-domain blocks. Each of these blocks is the same length of time. For example, one second, two seconds, 50 milliseconds, and the like. In addition, these blocks overlap equally so that half of each block (except the first and last) is duplicated in an adjacent block.

For example, suppose that an entire audio clip is divided into overlapping, two-second long, time-domain blocks. In this example, each block has a one second overlap with its neighbors. If the clip were about 3.5 minutes long, then there would be about 210 blocks.

Fig. 4A shows a graph 300 of an audio signal in the time domain. Time advances from left to right. Fig. 4B shows a graph 310 of the same audio signal sampled over the same time period. Fig. 4B includes a block 312 representing a first of equally spaced, overlapping, time-domain blocks.

Each block is transformed by a MCLT (modulated complex lapped transform) to the frequency domain. This produces a vector have a defined number of magnitude components. The magnitude is measured in a logarithmic scale, in decibels (dB).

Frames. Fig. 4C shows a graph 320 of the same audio signal sampled over the same time period. In Fig. 4C, there is a set 330 of five adjacent blocks 332-339. The blocks represent equally spaced, overlapping, time-domain blocks. (For simplicity, the overlapping nature of the blocks is not shown.) The set 330 of blocks is called a "frame."

In general, a frame may include any given number of blocks. However, if it too long, the watermark is more likely to be noticed by a digital pirate. If it is too short, the bits of the watermark may be had to find for the watermark detector. In addition, the optimum number of blocks in a frame depends upon the block size. The proper number of block per frames for a given implementation can be determined with a minimum of empirical measurements. Three to seven blocks per frame may be appropriate for one implementation, but nine to eleven blocks per frame may be better for another.

Bitframes. Fig. 4D shows a graph 340 of the same audio signal sampled over the same time period. In Fig. 4D, there is a series of three contiguous frames 342, 344, and 346. Each frame has five neighboring blocks. (For simplicity, the overlapping nature of the blocks is not shown.) The series of contiguous frames (342, 344, and 346) is called a "bitframe."

In general, a bitframe may include any given number of frames. One bit of the covert data is encoded into each bitframe. Hence, the name "bitframe." The proper number of frames per bitframe for a given implementation can be determined with a minimum of empirical measurements. Four frames per bitframe may be appropriate for one implementation, but eight frames per bitframe may be better for another.

Windows. Fig. 4E shows a graph 350 of the same audio signal sampled over the same time period. In Fig. 4E, there is a collection 370 of neighboring bitframes 360 and 362. Each bitframe is composed of five contiguous frames (such as frame 354). Each frame has five adjacent blocks (such as block 352). (For simplicity, the overlapping nature of the blocks is not shown.) The collection of neighboring bitframes (360 and 362) is called a "window."

In general, a window may include any given number of bitframes. The proper number for a given implementation can be determined with a minimum of empirical measurements. Four bitframes per window may be appropriate for one implementation, but ten bitframes per window may be better for another.

Graph illustrating Blocks, Frames, Bitframes, and Windows. Fig. 5 shows a graph 400 of the same audio clip of Figs. 4A-4E, but this graph does not show the clip in the time domain. Rather, it shows a graph in the frequency-domain for each overlapping, time-domain block (like blocks 332-339 in Fig. 4C). Time advances from left to right. Frequency increases from bottom to top. From zero to a maximum frequency of interest ("MaxFreq").

In Fig. 5, each of blocks 412a-g contain a frequency-domain graph for its time blocks. Horizontal hash marks, like mark 414, represent the magnitude of a given frequency range. The watermark is encoded in multiple frequency subbands in a range from "$SubBand_{max}$" line and "$SubBand_{min}$" line as shown in Fig. 5.

A given number of blocks (such as blocks 412a-g) form a "frame" (such as frame 410). Each frame in this audio clip includes the same number of blocks. In Fig. 5, frames 420 and 430 include the same number of blocks.

Fig. 5 also illustrates bitframes 460, 470, and 480 and a window 490. Bitframe 460 includes three frames (410, 420, and 430). Likewise, each bitframe in this clip contains three frames.

Encoding Bits of a Watermark

A watermark is composed of a given number of bits (such as eighty bits). The bits of a watermark are encoded by slightly increasing and decreasing the magnitude of frequencies within a block. More specifically, magnitudes of subbands of frequencies are slightly modified.

This slight change is plus or minus Q decibel (dB), where Q is set to 1 for example. Overall, these frequency changes are not heard because they are so tiny. Again, these frequency magnitudes are represented by horizontal hash marks in Fig. 5, like mark 414.

More specifically, only the frequencies between the $SubBand_{max}$ and $SubBand_{min}$ lines are modified to encode a bit of the watermark.

Redundancy Encoding

Successive Redundancy of Full Watermark. Using the exemplary watermarking, successive bits of a watermark are encoded into successive frames. One bit is encoded in each frame. For example, suppose the watermark is eighty bits long. The first three bits of the watermark in this example is "101." Also, suppose that frame 410 is frame one, frame 420 is frame two, and so forth until frame eighty is reached.

In this example, frame 410 of Fig. 5 will have the first bit of the watermark encoded therein. That bit is "1" and is represented by circular "bit-value-indicator" 450. Frame 420 will have the second bit of the watermark encoded therein. That bit is "0" and is represented by bit-value-indicator 452. Frame 430 will have the third bit of the eighty-bit watermark encoded therein. That bit is "1" and is represented by bit-value-indicator 454.

Typically, the full audio clip in which the watermark is being encoded is longer than time elapsed for the eighty frames. Therefore, this process is repeated until the end of the audio clip. In one implementation, it was determined that approximately eleven seconds was required to encode a watermark. Thus, in a four-minute clip, the watermark will be encoded approximately twenty-one times in successive sets of eighty frames.

Redundancy within a Frame. As described above, each frame has one bit of the watermark encoded therein. That one bit is encoded in each block of a frame. This means that within each block in a frame is encoded the exact same bit. For example, bit-value-indicator 450 of Fig. 5 shows that each block in frame 410 has a bit value of "1" encoded therein.

When a bit of a watermark is detected from an audio clip, the detector reads the bit from the block in the middle of frame. In frame 410 of Fig. 5, the middle block is block 412d.

The redundancy within a frame is designed to thwart malevolent desynchronization attacks in the time-domain. In other words, it lessens the effect of time-shifting the audio clip. Since it reads what it believes to be the middle block of a frame, it will still read the correct bit value even if the clips is shifted over an amount of time equal to about half of a frame.

Covert Channel

A covert data channel (or simply "covert channel") is a hidden layer of data imposed upon an existing data channel. In other words, a data signal containing a given amount of information data is modified so that it carries additional information data. Furthermore, this additional information layer is added without increasing the bandwidth to carry the signal. For example, suppose that a digital signal carries a hundred bits of data with the original information data encoded therein. With the covert channel, the new signal still carries a hundred bits.

Moreover, the additional information data of the covert channel is not readily identifiable as such. Rather, to an uninformed observer, the signal appears to convey only one layer of information.

Covert Channel Watermarking

There are two exemplary embodiments of the exemplary covert channel watermarking. One is "modulated covert channel" and the other is "multiple watermarks covert channel." Both layer a covert message over a watermark. The resulting coverted watermark is encoded into a signal. A covert message may be nearly any type of message. For example, a covert message may be a street address, a phone number, a name, a Web address, an e-mail address, terms of a license, etc.

Both embodiments clandestinely insert a covert message over the existing watermark and the watermark is encoded into a digital audio signal. Both impose one bit of the covert message over an entire bitframe. Hence, the name "bitframe." A bitframe, itself, encodes some portion of the watermark. That portion is equivalent to X number of bits of the watermark, where X is the number of frames in a bitframe. Herein, "imposing" a covert data channel over an original data channel (i.e., one or more bits of a data pattern) means that the original data channel

is altered or modified to carry the covert channel. However, the size (i.e., number of bits, bandwidth) of the data channel encoded in a signal is no greater than the original data channel. Furthermore, it means that the covert data channel is decipherable and the existing data channel is, at least, detectible. Herein, "extracting" is the reverse of imposing. The covert data channel is decipherable within the data channel of the signal.

Fig. 5 shows bitframes 460, 470, 480. It also shows each bitframe with a diamond-shaped bit-value-indicator (462, 472, and 482). Each bit-value-indicator specifies the value of the bit of the covert message (i.e., "covert bit") imposed upon all of the blocks and frames of the bitframe. For example, the value "0" of bit-value-indicator 462 of bitframe 460 is imposed upon all of the blocks (such as 412a-412g) of all of the frames (410, 420, and 430) of bitframe 460.

Unlike the watermark itself, the covert message need not repeat throughout the clip. It may repeat, but it need not repeat at the same frequency. A window (such as window 490 of Fig. 5) is a collection of bitframes that represents either the entire covert message or some portion of the covert message. The next window may include a different covert message or the next portion of such message. Alternatively, the next window may repeat the same message.

The combining of a covert bit over a chip of the watermark is accomplished, for example, by XORing the covert bit with the bits of the chip together. The results of such operation are encoded into the signal.

XOR is a Boolean operator (also known as the "exclusive OR" operator) that returns a value of TRUE ("1") only if just one of its operands is TRUE ("1"). In contrast, an inclusive OR operator returns a value of TRUE ("1") if either or both of its operands are TRUE ("1").

Modulated Covert Channel. Each covert bit is modulated up or down to generate the covert message. More specifically, up indicates a "1" and down indicates a "0." More specifically still, the frequency magnitudes are modulated one of two discrete states: "+Q" for "1" and "–Q" for zero, where Q is usually set to 1.0 dB. The detector can determine the message by translating the positive and negative results (i.e., two states) into the covert message.

In order to detect the watermark with this modulated covert channel layered onto of it, the detector will focus on the absolute value of the results of some of its watermark detection processes (e.g., normalized correlation formulas). Otherwise, the results of such processes will produce negative values when they would have produced positive values absent such a modulated covert channel.

Using this modulated covert channel, the window size may be any specified size. It may span across multiple repetitions of the watermark.

Multiple Watermark Covert Channel. In this technique, a window is composed of N bitframes. Within each window, a N-bit chip of one of 2^N watermarks is encoded. That N-bit chip of one of 2^N watermarks indicates a value of an N-bit chip of a covert message. The covert message is the combination of these values of N-bit chips.

The pattern generator of the encoder generates 2^N watermarks based upon a given key. If N is four, then the encoder generates sixteen watermarks. The encoder selectively encodes a chip of one of the sixteen watermarks into each window. The window is four (where N=four) bitframes in size; thus, it can encode a 4-bit chip therein.

The table below illustrates the relationship between each of sixteen watermarks (where N=4; thus 2^N=16) and each chip of four bits (where N=4) of a covert message:

Chip of Covert Message	2^4 Watermarks
0000	watermark 1
0001	watermark 2
0010	watermark 3
0011	watermark 4
0100	watermark 5
0101	watermark 6
0110	watermark 7
0111	watermark 8
1000	watermark 9
1001	watermark 10
1010	watermark 11
1011	watermark 12
1100	watermark 13
1101	watermark 14
1110	watermark 15
1111	watermark 16

When the chip of the covert message is "1101," then "watermark 14" is encoded within the signal. Conversely, when the detector detects "watermark 14," it knows that the chip of the covert message is "1101." The detector can recognize the detected watermark to be one the sixteen watermarks because its pattern

generator generates the sixteen watermarks based upon the same key as the encoder used.

Permutations

Herein, a "permutation" is a reordering of a set of data and a "reverse of a permutation" is returning a permuted set of data back to its original order. A collection of bits (such as a chip in a watermark) is an example of a set of data. Permutation may be used to further hide covert bits, which are written to the bitframes of a window. Alternatively, it may be used to further hide the watermark alone (without any covert channel).

Fig. 6 shows a graph 600 of an audio clip, but this graph does not show the clip in the time domain. Rather, it shows a graph in the frequency-domain of bitframes of frames of overlapping, time-domain blocks. For simplicity, the frames and blocks are not shown. Time advances from left to right. Frequency increases from bottom to top.

Graph 600 shows an example window 630 of four bitframes 612, 614, 616, and 618. The bitframes have pentagon-shaped value-indicators (622, 624, 626, and 628). These value-indicators include value-labels (b0, b1, b2, and b3). Each label represents the values of the bits inserted into a given bitframe. More specifically, the label represents the values ultimately encoded into a given bitframe of a signal. Such values may be a chip of a watermark alone or it may be the values of a chip of a watermark with a covert channel layered thereon.

In their original order, b0 is encoded into bitframe 612, b1 is encoded into bitframe 614, b2 is encoded into bitframe 616, and b3 is encoded into bitframe 618. Without permutation, the values of these bitframes are encoded and detected in the

order of advancing time (from left to right on the graph). In this case, the order is "b0, b1, b2, b3."

The re-arrangement of the original order "b0, b1, b2, b3" is a permutation of the bitframes in the window 630. For example, the original order may be permuted into this order "b1, b2, b0, b3." In this example, b1 is encoded into bitframe 612, b2 is encoded into bitframe 614, b0 is encoded into bitframe 616, and b3 is encoded into bitframe 618.

The encoder bases its permutations upon a permutation table. This table may be precreated and stored at the encoder. This table may be pseudorandomly generated based upon a given key.

When the detector receives a signal with a permuted watermark, it uses the same permutation table (that the encoder used) to reverse permute the permuted watermark. Therefore, either the detector has the same table stored therein or it pseudorandomly generates it in the same manner and using the same key as the encoder.

Rather than re-order (i.e., permute) the entire audible spectrum (e.g., between $SubBand_{min}$ and $SubBand_{max}$), the exemplary watermarking with permutations only permutes selected subbands of frequencies within this spectrum. Fig. 6 shows four of such subbands 642, 644, 646, and 648.

The audible spectrum is divided into multiple frequency subbands. When permuting a bitframe, only selected subbands (such as subbands 642, 644, 646, and 648) are permuted. The other subbands (between and around subbands 642, 644, 646, and 648) in the spectrum are not permuted.

Since the permutation applies to only selected subbands in the spectrum, this permutation further hides the watermark data (with/without covert channel) within

the signal. In effect, the permutation "mixes-up" the data to make a pattern even more difficult to detect. However, the watermark detector can locate the original signal, in part, because it has the permutation table.

The permutation of covert bits within selected subbands enables much better security and noise distribution. The covert message and the watermark are harder to find. The "noise" introduced by the covert message and the watermark is more evenly distributed; thereby, reducing the affect on the audio quality.

Methodological Implementation of

Exemplary Watermark Encoding with Covert Channel and Permutations

Fig. 7 shows a methodological implementation of the exemplary watermark encoding with covert channel and permutations. At 650, an original audio signal (such as from an audio clip) is preprocessed. One effective result of such preprocessing is to produce blocks and frames as described above.

Furthermore, such signal preprocessing is generally described above in reference to the watermark encoding system of Fig. 2. It is also described in more detail in co-pending patent application: U.S. Patent Application Serial No. 09/316,899, entitled "Audio Watermarking with Dual Watermarks" filed on May 22, 1999.

At 652, the watermark encoder generates a watermark in accordance with the watermark generation described above and in the "Audio Watermarking with Dual Watermarks" co-pending application.

At 654, the blocks of the audio signal, the watermark, a covert message, and a permutation table are provided to a watermark insertion unit (such as unit 116 in Fig. 2). At 656, before the covert message is combined with the watermark, the

covert message is permuted a window at a time. For example, a permutation subunit (such as PSU 124 in Fig. 2) uses a permutation table to determine how to reorder values in a window.

At 658 in Fig. 7, the covert channel is embedded into the watermark. For example, a covert message subunit (such as CMSU 122 of Fig. 2) imposes the covert message onto the watermark. Alternatively, first, the covet message may be imposed upon the watermark and second, a window of the watermark (with the covert channel) is permuted.

At 660, the resulting watermark is inserted into the audio signal. At 662, this process ends.

The following is an example of pseudocode that may be used to implement exemplary watermark encoding with covert channel and permutations:

```
BALANCING NOISE INDUCED BY THE CARRIER OF THE WATERMARK USING SECRET
   PERMUTATIONS OF MODULATED BITS OF THE COVERT COMMUNICATION CHANNEL
-------------------------------------------------------------
INPUT=WATERMARK(SUBBANDS S, BITFRAMES B, FRAMES F),
   COVERT_CHANNEL_BITS B, {SECRET_KEY}
OUTPUT=SECRETLY PERMUTED WATERMARK spWATERMARK(SUBBANDS S, BITFRAMES
   B, FRAMES F)
-------------------------------------------------------------
COVERT CHANNEL IS MODULATED INTO THE WATERMARK BY
FOREACH BITFRAME b in F
   newWATERMARK(SUBBANDS S, BITFRAME b, FRAMES F) =
   WATERMARK((SUBBANDS S, BITFRAME b, FRAMES F) xor
   COVERT_CHANNEL_BITS(BIT b)
ENDFOREACH
SECRETLY PERMUTED WATERMARK spWATERMARK(SUBBANDS S, BITFRAMES B,
   FRAMES F)
is created by permuting the bits of the modulated COVERT_CHANNEL_BITS
   for each subband separately.
FOREACH SUBBAND s in S
   spCOVERT_CHANNEL_BITS(B) = create secret permutation of bits of
   COVERT_CHANNEL_BITS(B)
   FOREACH BITFRAME b in F
         newWATERMARK(SUBBAND s, BITFRAME b, FRAMES F) =
   WATERMARK((SUBBAND s, BITFRAME b, FRAMES F) xor
   spCOVERT_CHANNEL_BITS(BIT b)
ENDFOREACH
=============================================================
=============================================================
```

Methodological Implementation of

Exemplary Watermark Detecting with Covert Channel and Permutations

Fig. 8 shows a methodological implementation of the exemplary watermark detecting with covert channel and permutations. At 680, a watermarked audio signal (such as from an audio clip) is preprocessed. The watermark is permuted and includes a covert channel. The effective result of such preprocessing is to produce blocks and frames.

Furthermore, such signal preprocessing is generally described above in reference to the watermark detecting system of Fig. 3. It is also described in more detail in co-pending patent application: U.S. Patent Application Serial No. 09/316,899, entitled "Audio Watermarking with Dual Watermarks" filed on May 22, 1999.

At 682, the watermark detector generates a comparison watermark in accordance with watermark generation described above and in the "Audio Watermarking with Dual Watermarks" co-pending application. This comparison watermark is generated using the same key as the original watermark. Therefore, they are identical.

At 684, the blocks of the audio signal, the comparison watermark, and the permutation table are provided to a watermark detector unit (such as unit 146 in Fig. 3). At 686, the watermark is detected from the audio signal and compared to the comparison watermark.

At 686 in Fig. 8, the permutation is reversed. For example, a permutation subunit (such as PSU 154 in Fig. 3) uses a permutation table to determine how to reverse the permutation values in a window and restore it to its original order. In addition, the covert message is extracted from the watermark. For example, a

covert message subunit (such as CMSU 152 of Fig. 3) extracts (i.e., interprets) the covert message from the watermark.

At 690, this process ends. Typically, the detector will generate a result that indicates whether a watermark is present in the audio signal.

Exemplary Computing Environment

Fig. 9 illustrates an example of a suitable computing environment 920 on which the exemplary watermarking may be implemented.

Exemplary computing environment 920 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the exemplary watermarking. Neither should the computing environment 920 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary computing environment 920.

The exemplary watermarking is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the exemplary watermarking include, but are not limited to, personal computers, server computers, think clients, thick clients, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The exemplary watermarking may be described in the general context of computer-executable instructions, such as program modules, being executed by a

computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The exemplary watermarking may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

As shown in Fig. 9, the computing environment 920 includes a general-purpose computing device in the form of a computer 930. The components of computer 920 may include, by are not limited to, one or more processors or processing units 932, a system memory 934, and a bus 936 that couples various system components including the system memory 934 to the processor 932.

Bus 936 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnects (PCI) buss also known as Mezzanine bus.

Computer 930 typically includes a variety of computer readable media. Such media may be any available media that is accessible by computer 930, and it includes both volatile and non-volatile media, removable and non-removable media.

In Fig. 9, the system memory includes computer readable media in the form of volatile, such as random access memory (RAM) 940, and/or non-volatile memory, such as read only memory (ROM) 938. A basic input/output system

(BIOS) 942, containing the basic routines that help to transfer information between elements within computer 930, such as during start-up, is stored in ROM 938. RAM 940 typically contains data and/or program modules that are immediately accessible to and/or presently be operated on by processor 932.

Computer 930 may further include other removable/non-removable, volatile/non-volatile computer storage media. By way of example only, Fig. 9 illustrates a hard disk drive 944 for reading from and writing to a non-removable, non-volatile magnetic media (not shown and typically called a "hard drive"), a magnetic disk drive 946 for reading from and writing to a removable, non-volatile magnetic disk 948 (e.g., a "floppy disk"), and an optical disk drive 950 for reading from or writing to a removable, non-volatile optical disk 952 such as a CD-ROM, DVD-ROM or other optical media. The hard disk drive 944, magnetic disk drive 946, and optical disk drive 950 are each connected to bus 936 by one or more interfaces 954.

The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules, and other data for computer 930. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 948 and a removable optical disk 952, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 948, optical disk 952, ROM 938, or RAM 940, including, by way of example, and

not limitation, an operating system 958, one or more application programs 960, other program modules 962, and program data 964.

A user may enter commands and information into computer 930 through input devices such as keyboard 966 and pointing device 968 (such as a "mouse"). Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, serial port, scanner, or the like. These and other input devices are connected to the processing unit 932 through an user input interface 970 that is coupled to bus 936, but may be connected by other interface and bus structures, such as a parallel port, game port, or a universal serial bus (USB).

A monitor 972 or other type of display device is also connected to bus 936 via an interface, such as a video adapter 974. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers, which may be connected through output peripheral interface 975.

Computer 930 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 982. Remote computer 982 may include many or all of the elements and features described herein relative to computer 930.

Logical connections shown in Fig. 9 are a local area network (LAN) 977 and a general wide area network (WAN) 979. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When used in a LAN networking environment, the computer 930 is connected to LAN 977 network interface or adapter 986. When used in a WAN networking environment, the computer typically includes a modem 978 or other means for establishing communications over the WAN 979. The modem 978,

which may be internal or external, may be connected to the system bus 936 via the user input interface 970, or other appropriate mechanism.

Depicted in Fig. 9, is a specific implementation of a WAN via the Internet. Over the Internet, computer 930 typically includes a modem 978 or other means for establishing communications over the Internet 980. Modem 978, which may be internal or external, is connected to bus 936 via interface 970.

In a networked environment, program modules depicted relative to the personal computer 930, or portions thereof, may be stored in a remote memory storage device. By way of example, and not limitation, Fig. 9 illustrates remote application programs 989 as residing on a memory device of remote computer 982. It will be appreciated that the network connections shown and described are exemplary and other means of establishing a communications link between the computers may be used.

Exemplary Operating Environment

Fig. 9 illustrates an example of a suitable operating environment 920 in which the exemplary watermarking may be implemented. Specifically, the exemplary watermarking is implemented by any program 960-962 or operating system 958 in Fig. 9.

The operating environment is only an example of a suitable operating environment and is not intended to suggest any limitation as to the scope of use of functionality of the exemplary watermarking described herein. Other well known computing systems, environments, and/or configurations that may be suitable for use with the exemplary watermarking include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, programmable consumer electronics, wireless

communications equipment, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

Computer-Executable Instructions

An implementation of the exemplary watermarking may be described in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically, the functionality of the program modules may be combined or distributed as desired in various embodiments.

Computer Readable Media

An implementation of the exemplary watermarking may be stored on or transmitted across some form of computer readable media. Computer readable media can be any available media that can be accessed by a computer. By way of example, and not limitation, computer readable media may comprise computer storage media and communications media.

Computer storage media include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium

which can be used to store the desired information and which can be accessed by a computer.

Communication media typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal such as carrier wave or other transport mechanism and included any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above are also included within the scope of computer readable media.

Conclusion

Although the improved audio watermarking with covert channel and permutations has been described in language specific to structural features and/or methodological steps, it is to be understood that the improved audio watermarking with covert channel and permutations defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed improved audio watermarking with covert channel and permutations.

CLAIMS

1. A method for concealing data within a digital signal, the method comprising:

receiving a first data pattern of discrete values and a second data pattern of discrete values;

imposing a discrete value of the second data pattern over one or more values of the first data pattern.

2. A method as recited in claim 1 further comprising encoding a third data pattern into the digital signal, wherein such third data pattern is the result of the imposing.

3. A method as recited in claim 1, wherein the imposing comprises performing a Boolean operation with a discrete value of the second data pattern and one or more values of the first data pattern.

4. A method as recited in claim 1, wherein the imposing comprises XORing a discrete value of the second data pattern with one or more values of the first data pattern.

5. A method as recited in claim 1, wherein

a pattern of discrete values may be encoded into the signal in one of multiple discrete states;

the imposing comprises encoding one or more values of the first data pattern into the digital signal into a state that indicates a discrete value of the second data pattern.

6. A method as recited in claim 1, wherein the digital signal is an digital audio signal.

7. A method as recited in claim 1, wherein the first data pattern is a watermark.

8. A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 1.

9. A method for revealing a covert data pattern of discrete values from an encoded data pattern of discrete values in a digital signal, the method comprising:

receiving the encoded data pattern;

extracting a discrete value of the covert data pattern from one or more values of the encoded data pattern.

10. A method as recited in claim 9 further comprising detecting an original data pattern within the encoded data pattern of the digital signal.

11. A method as recited in claim 9, wherein

a pattern of discrete values may be encoded into the signal in one of multiple discrete states;

the extracting comprises decoding a discrete value of the covert data pattern from the digital signal based upon a state of a one or more discrete values of the encoded data pattern.

12. A method as recited in claim 9, wherein the digital signal is an digital audio signal.

13. A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 9.

14. A method for encoding a watermark with a covert message into a digital audio signal, wherein binary bits of the watermark may be encoded into the signal in multiple states, the method comprising encoding one or more bits of the watermark into the digital signal into a state that indicates a discrete value of the covert message.

15. A method as recited in claim 14, wherein the multiple states are positive or negative modifications to magnitudes of one or more subbands in the frequency spectrum of a sample of the signal.

16. A method for imposing a covert message into a watermark, the method comprising:

generating multiple watermarks;

assigning a watermark to each of possible discrete value for a portion of the covert message;

selecting a watermark that corresponds to an actual discrete value of a specific portion of the covert message;

encoding the selected watermark into the signal.

17. A method as recited in claim 16, wherein

size of all portions of the covert message is N bits long;

quantity of the multiple watermarks is 2^N.

18. A computer-readable medium having computer-executable instructions that, when executed by a computer, perform a method for concealing data within a digital signal, the method comprising:

receiving a first data pattern of discrete values and a second data pattern of discrete values;

imposing a discrete value of the second data pattern over one or more values of the first data pattern.

19. A computer-readable medium having computer-executable instructions that, when executed by a computer, perform a method for revealing a covert data pattern of discrete values from an encoded data pattern of discrete values in a digital signal, the method comprising:

receiving the encoded data pattern;

extracting a discrete value of the covert data pattern from one or more values of the encoded data pattern.

20. An apparatus comprising:

a processor;

a covert-channel-encoder executable on the processor to:

receive a first data pattern of discrete values and a second data pattern of discrete values;

impose a discrete value of the second data pattern over one or more values of the first data pattern;

encode result of such imposing into a digital signal.

21. An apparatus comprising:

a processor;

a covert-channel-decoder executable on the processor to:

receive a encoded data pattern within a digital signal;

extract a discrete value of a covert data pattern from one or more values of the encoded data pattern.

22. A data encoding system for concealing data within a digital signal, the system comprising:

a receiver for receiving a first data pattern of discrete values and a second data pattern of discrete values;

an imposer coupled to such receiver, the imposer for imposing a discrete value of the second data pattern over one or more values of the first data pattern;

an encoder coupled to the receiver and the imposer, the encoder for inserting within the digital signal results of the imposer's imposing a discrete value of the second data pattern over one or more values of the first data pattern.

23. An operating system comprising an encoding system as recited in claim 22.

24. A marked signal with an encoded data channel therein, wherein such encoded data channel has a covert data channel imposed therein, the marked signal generated in accordance with the following acts:

receiving an original data pattern of discrete values and a covert data pattern of discrete values;

imposing a discrete value of the covert data pattern over one or more values of the original data pattern;

encoding results of the imposing within an unmarked signal to produce the marked signal.

25. A marked signal as recited in claim 24, wherein the imposing comprises performing a Boolean operation with a discrete value of the second data pattern and one or more values of the first data pattern.

26. A marked signal as recited in claim 24, wherein the imposing comprises XORing a discrete value of the second data pattern with one or more values of the first data pattern.

27. A marked signal as recited in claim 24, wherein

a pattern of discrete values may be encoded into the signal in one of multiple discrete states;

the imposing comprises encoding one or more values of the first data pattern into the digital signal into a state that indicates a discrete value of the second data pattern.

28. A marked signal as recited in claim 24, wherein the digital signal is an digital audio signal.

29. A marked signal as recited in claim 24, wherein the original data pattern is a watermark.

30. A method for concealing data within a digital signal, the method comprising:

receiving a set of data having an original order;

permuting the set of data so that it is in a different order than the original;

encoding the permuted set of data into the digital signal.

31. A method as recited in claim 30, wherein the permuting utilizes a permutation table to determine the order in which to permute the set of data.

32. A method as recited in claim 30, where in the set of data is a portion of a watermark.

33. A computer-readable medium having computer-executable instructions that, when executed by a computer, perform a method for concealing data within a digital signal, the method comprising:

receiving a set of data having an original order;

permuting the set of data so that it is in a different order than the original;

encoding the permuted set of data into the digital signal.

34. A modulated signal with an permuted data channel encoded therein, the signal generated in accordance with the following acts:

receiving a set of data having an original order;

permuting the set of data so that it is in a different order than the original;

encoding the permuted set of data into a digital signal to produce the modulated signal with an permuted data channel encoded therein.

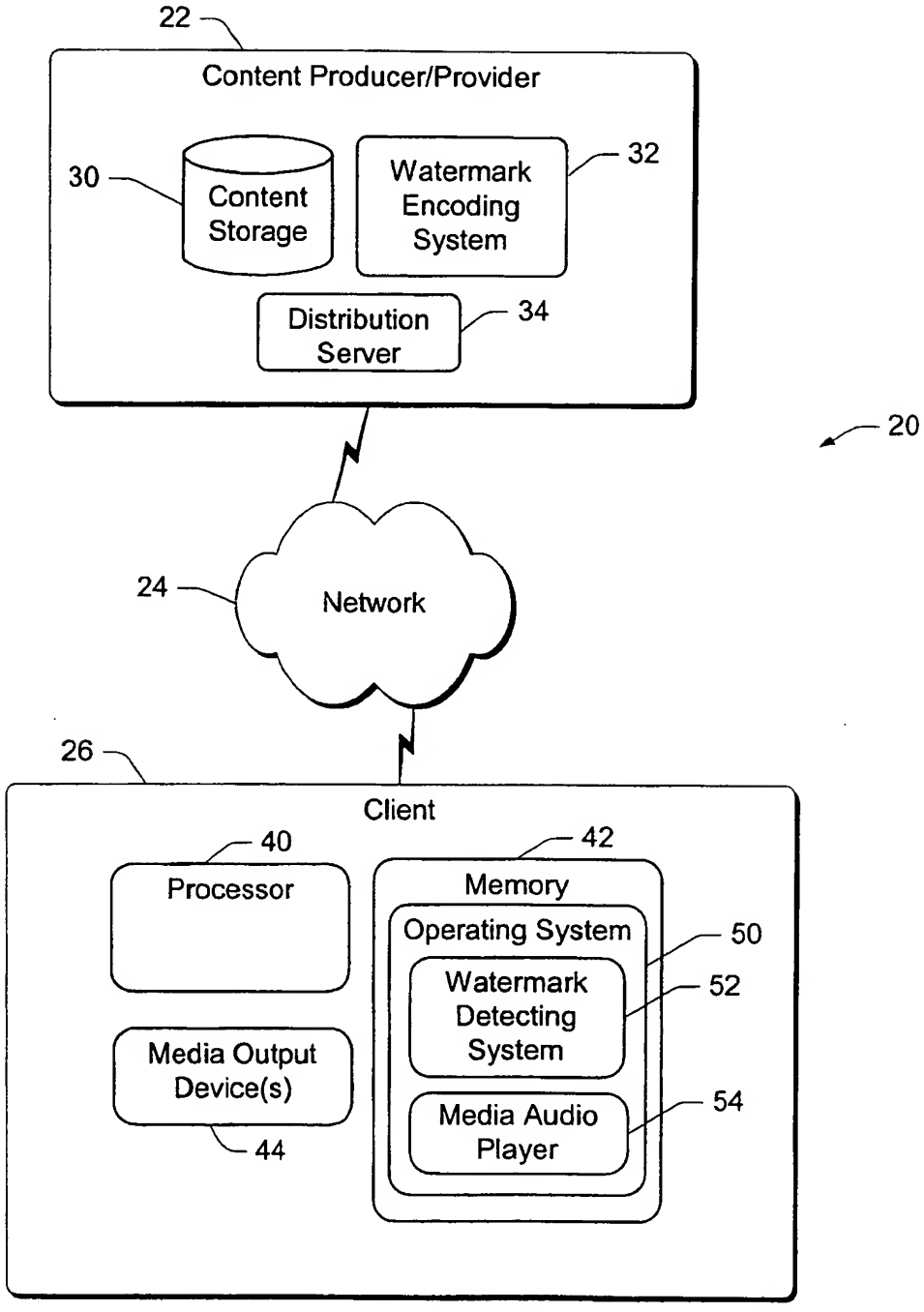

22
Content Producer/Provider
30
Content Storage
Watermark Encoding System
32
Distribution Server
34
20
24
Network
26
Client
40
Processor
42
Memory
Operating System
50
Watermark Detecting System
52
Media Output Device(s)
44
Media Audio Player
54

Fig. 1

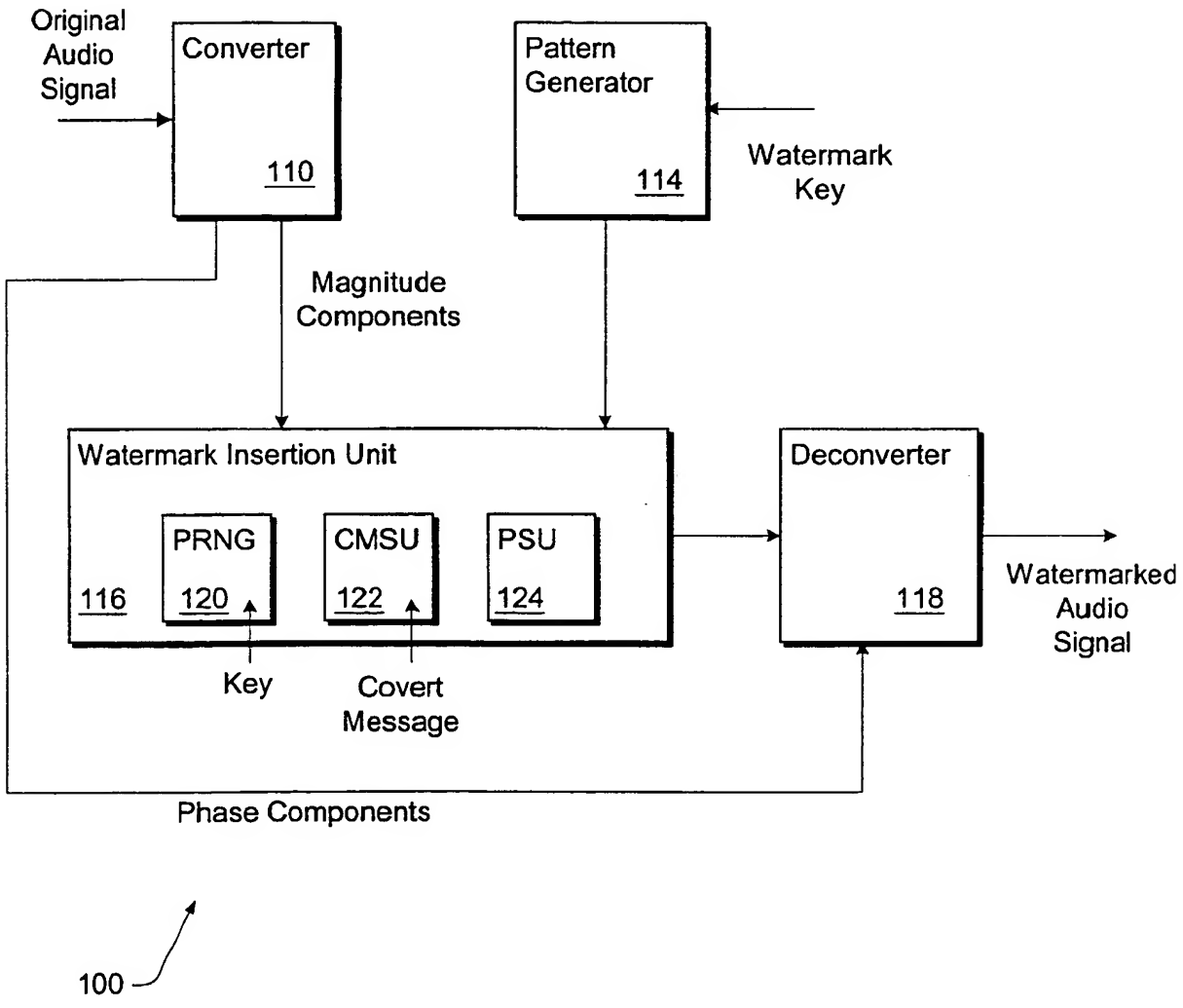

Original Audio Signal
Converter
110
Pattern Generator
114
Watermark Key
Magnitude Components
Watermark Insertion Unit
PRNG
120
CMSU
122
PSU
124
116
Deconverter
118
Watermarked Audio Signal
Key
Covert Message
Phase Components
100

Fig. 2

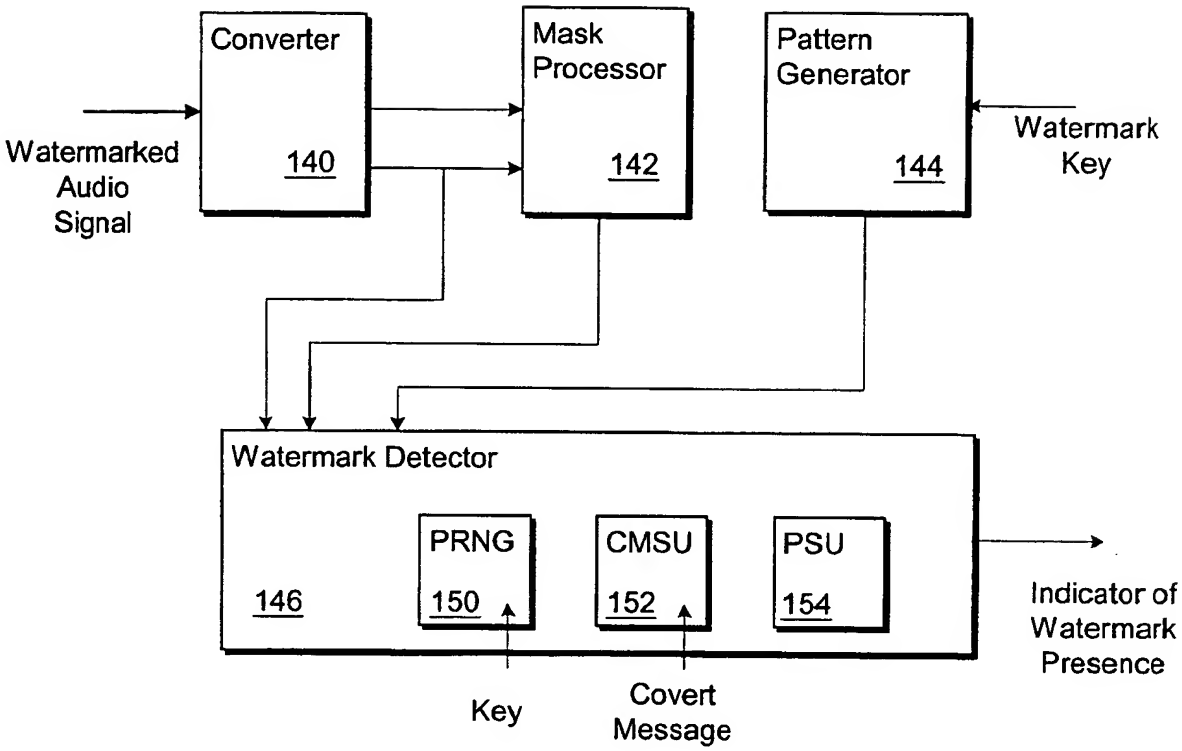

Converter
140
Mask Processor
142
Pattern Generator
144
Watermarked Audio Signal
Watermark Key
Watermark Detector
146
PRNG
150
CMSU
152
PSU
154
Indicator of Watermark Presence
Key
Covert Message
130

Fig. 3

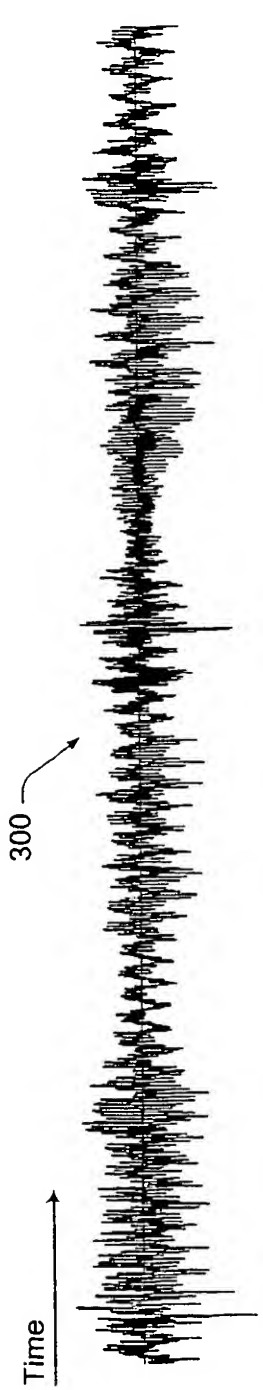


Fig. 4A

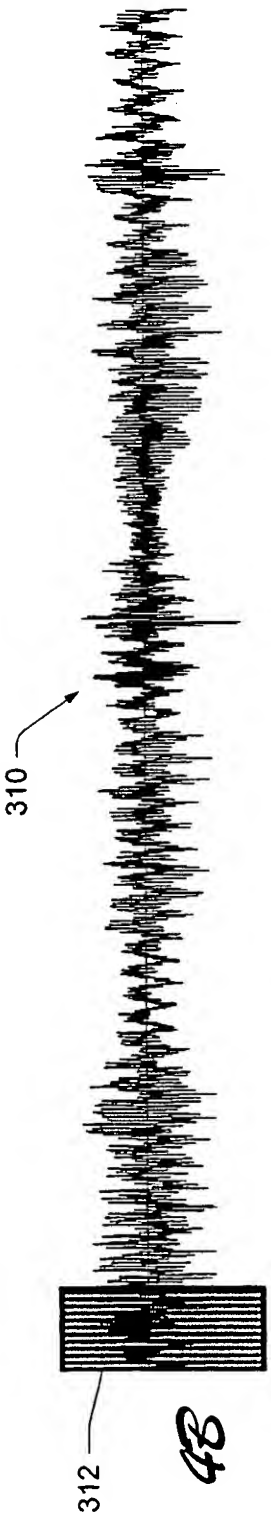


Fig. 4B

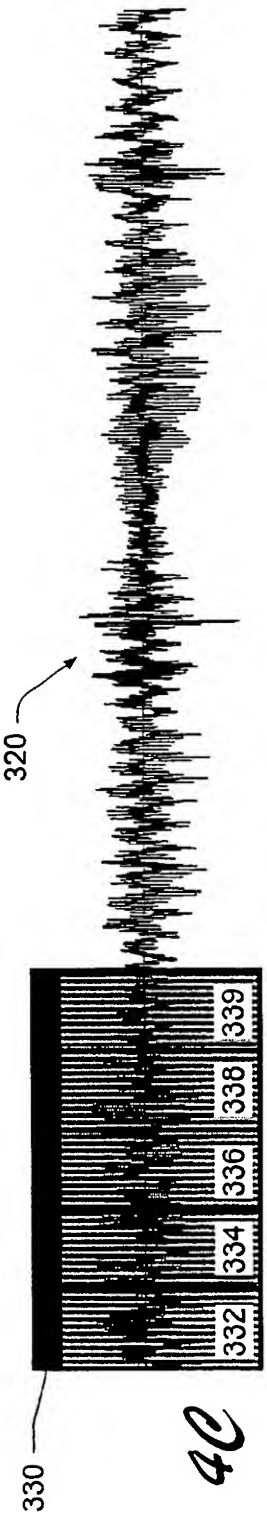


Fig. 4C

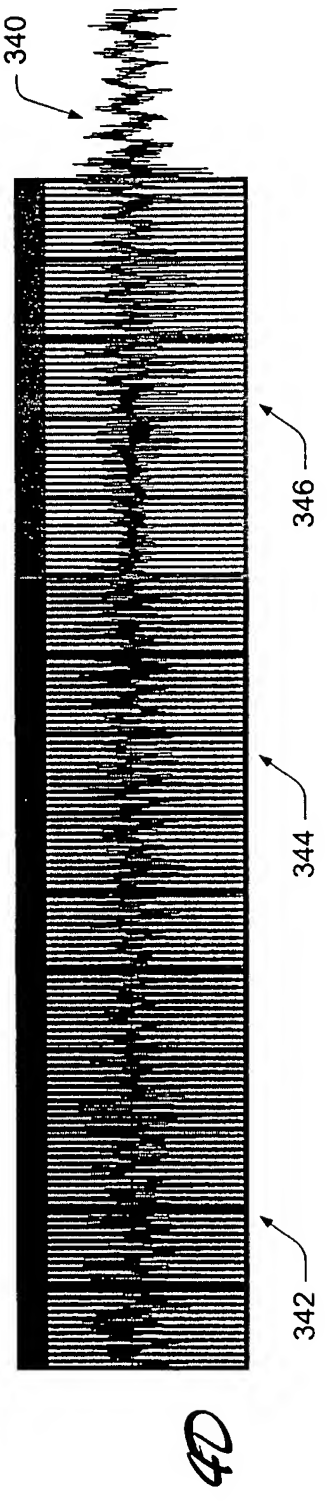


Fig. 4D

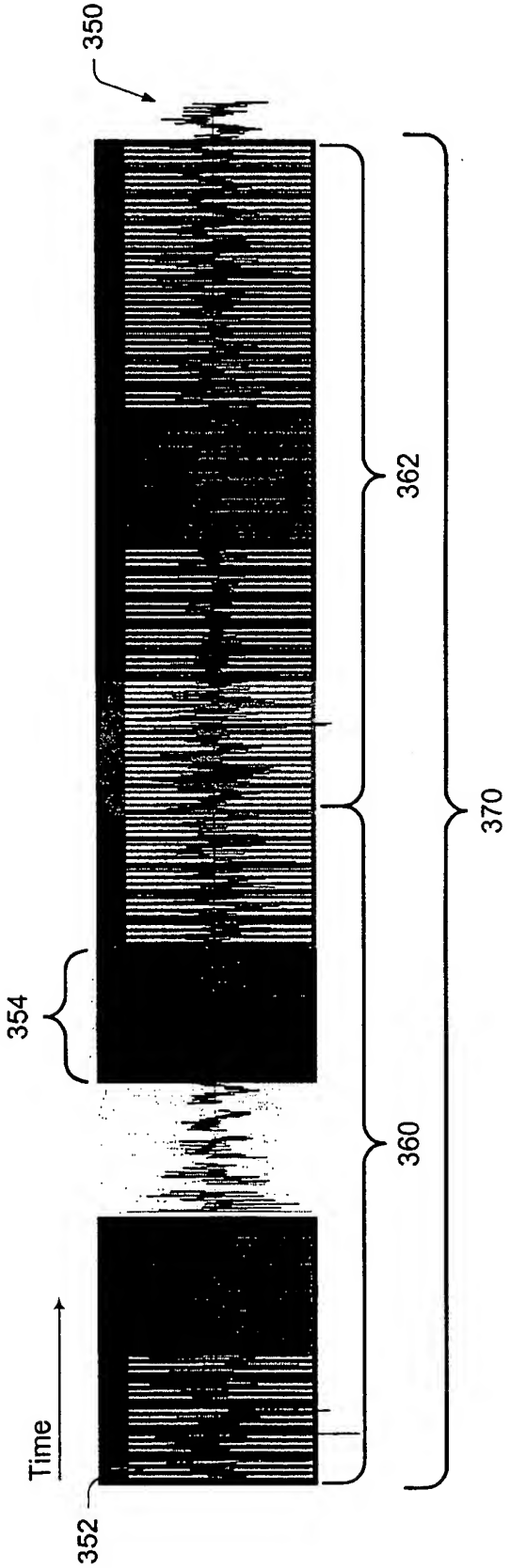


Fig. 4B

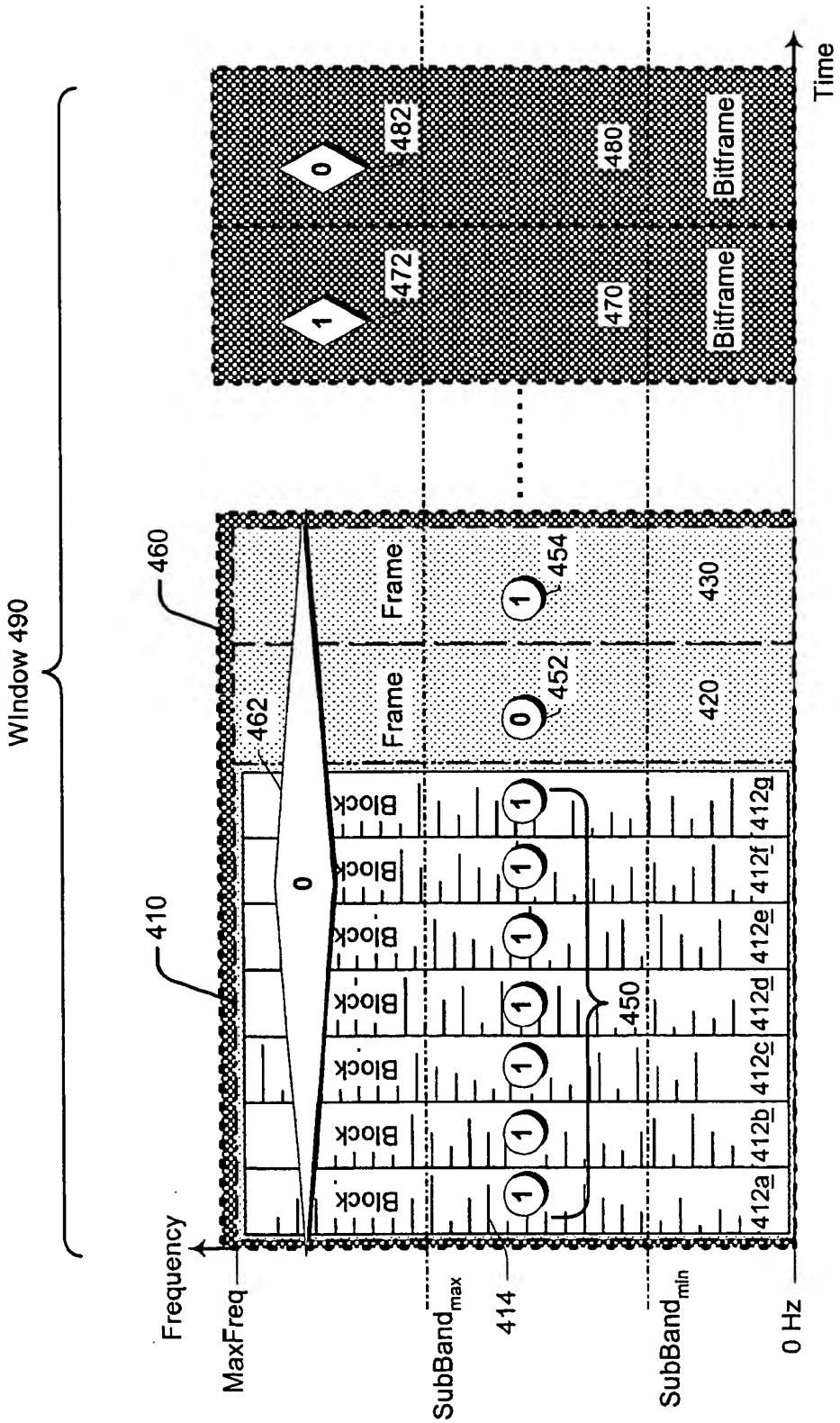

Window 490
460
410
462
Frame
Frame
Block
412a
412b
412c
412d
412e
412f
412g
0
1
452
454
450
414
420
430
470
472
480
482
Bitframe
Bitframe
Frequency
MaxFreq
SubBand$_{max}$
SubBand$_{min}$
0 Hz
Time
400

Fig. 5

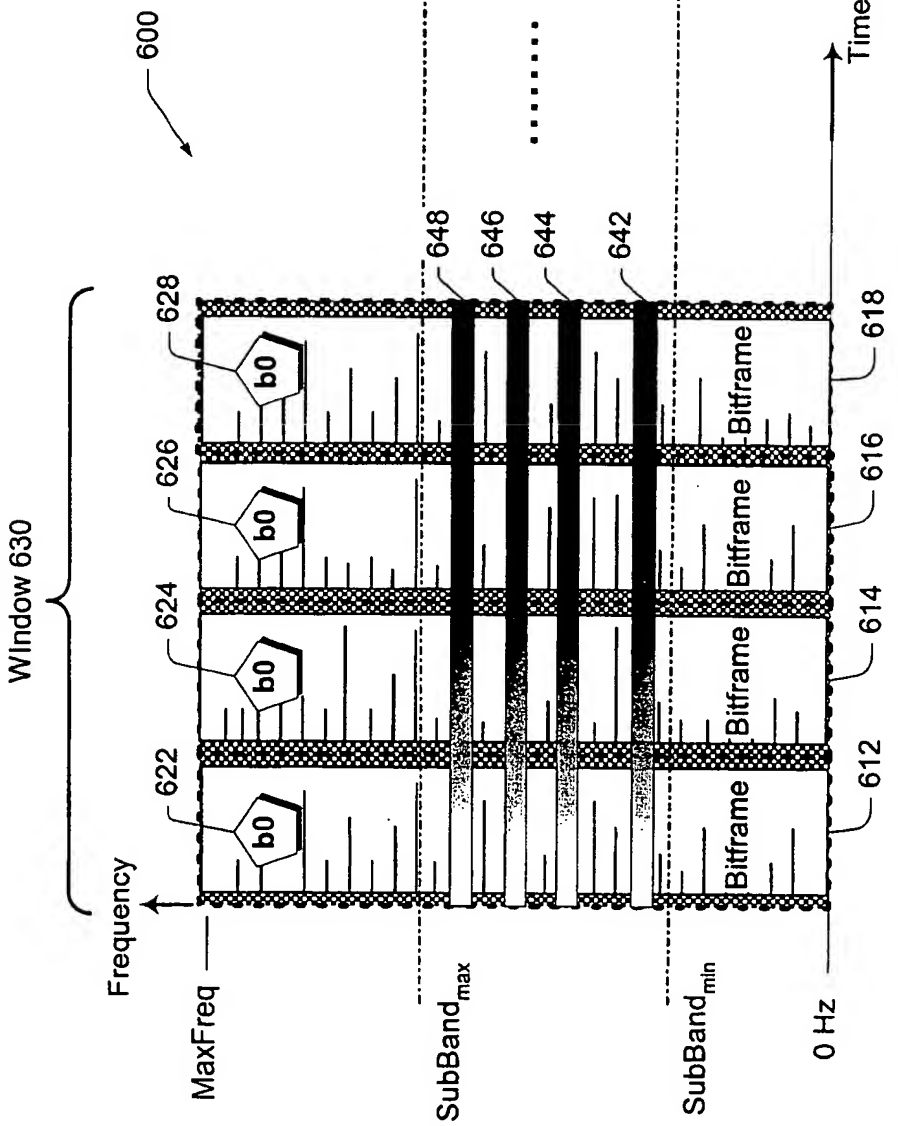

600
Window 630
Frequency
MaxFreq
SubBand$_{max}$
SubBand$_{min}$
0 Hz
Time
628
626
624
622
b0
b0
b0
b0
648
646
644
642
618
616
614
612
Bitframe
Bitframe
Bitframe
Bitframe

Fig. 6

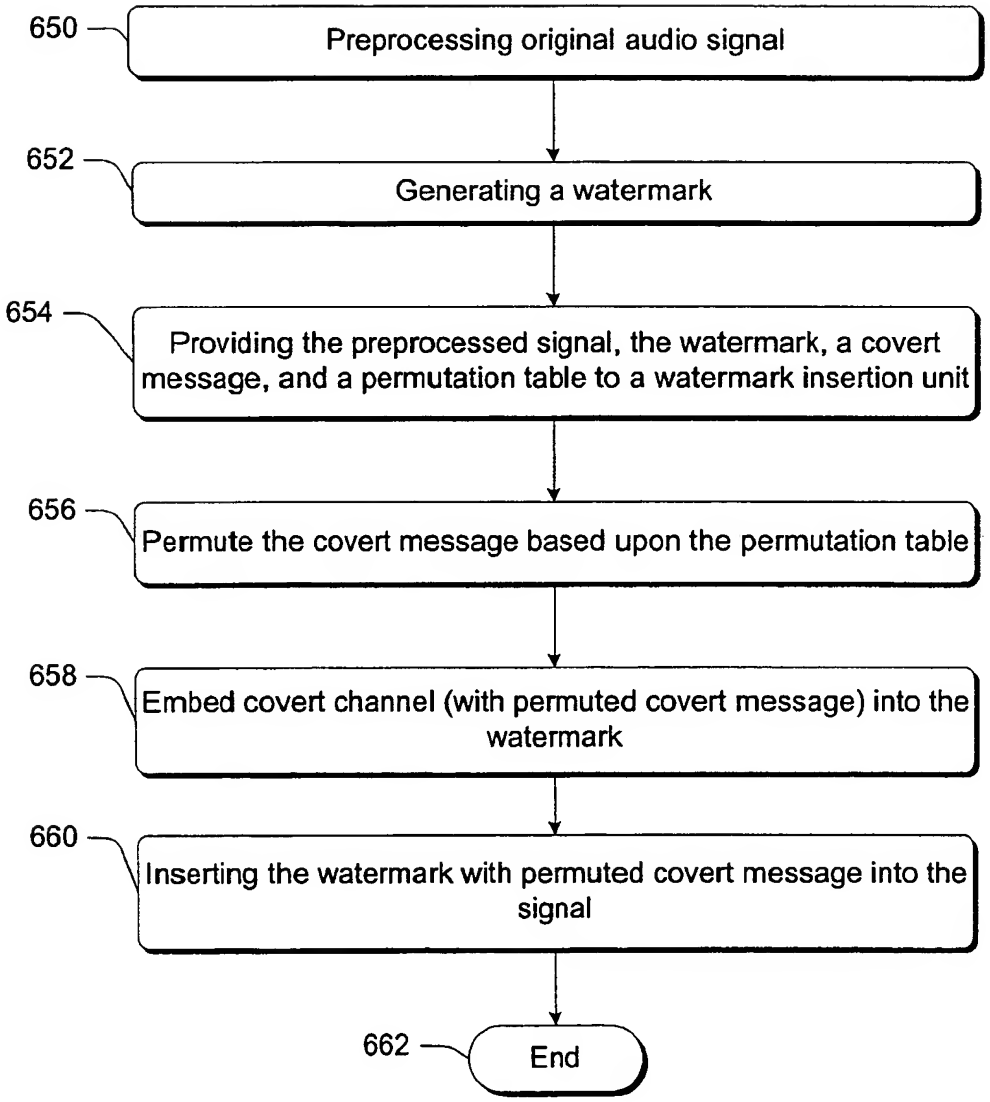

650
Preprocessing original audio signal
652
Generating a watermark
654
Providing the preprocessed signal, the watermark, a covert message, and a permutation table to a watermark insertion unit
656
Permute the covert message based upon the permutation table
658
Embed covert channel (with permuted covert message) into the watermark
660
Inserting the watermark with permuted covert message into the signal
662
End

Fig. 7

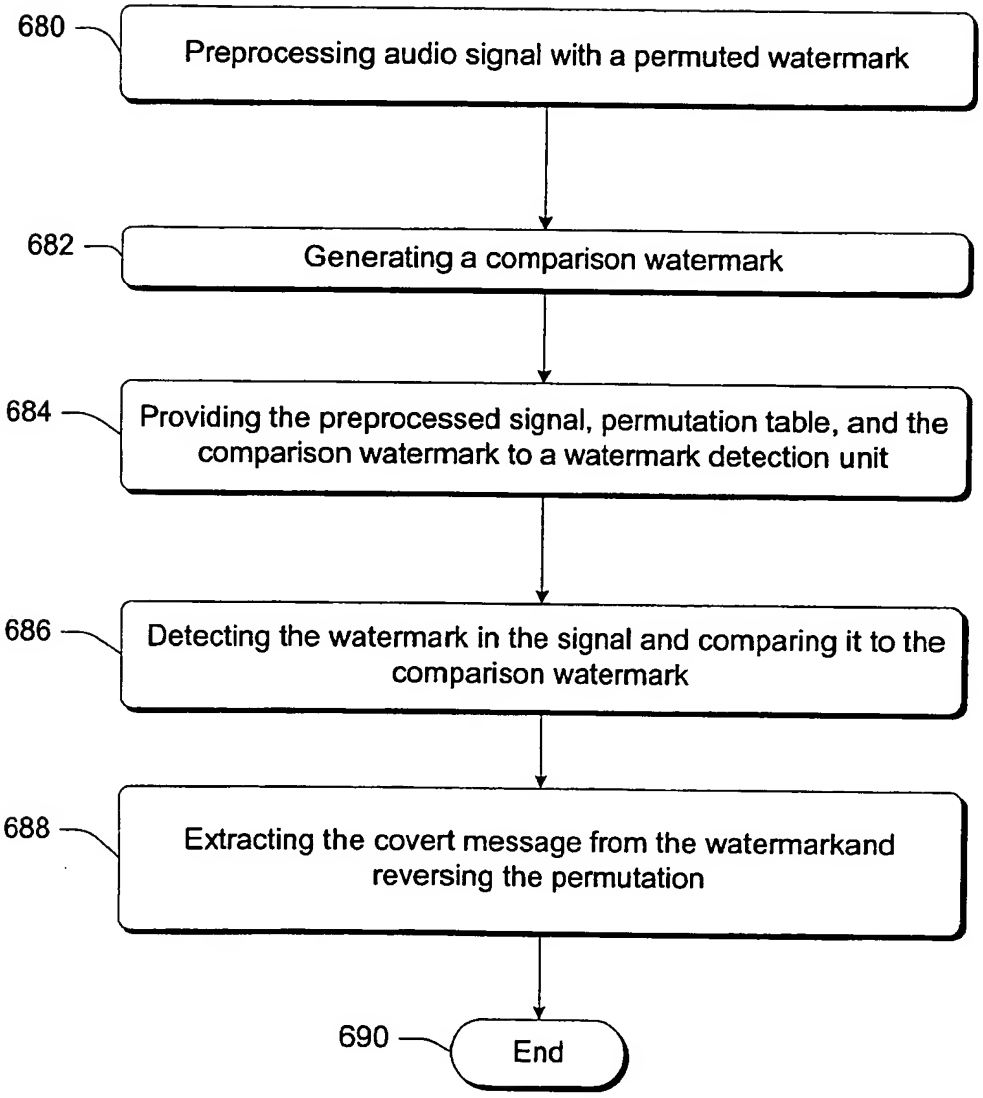

680
Preprocessing audio signal with a permuted watermark
682
Generating a comparison watermark
684
Providing the preprocessed signal, permutation table, and the comparison watermark to a watermark detection unit
686
Detecting the watermark in the signal and comparing it to the comparison watermark
688
Extracting the covert message from the watermarkand reversing the permutation
690
End

Fig. 8

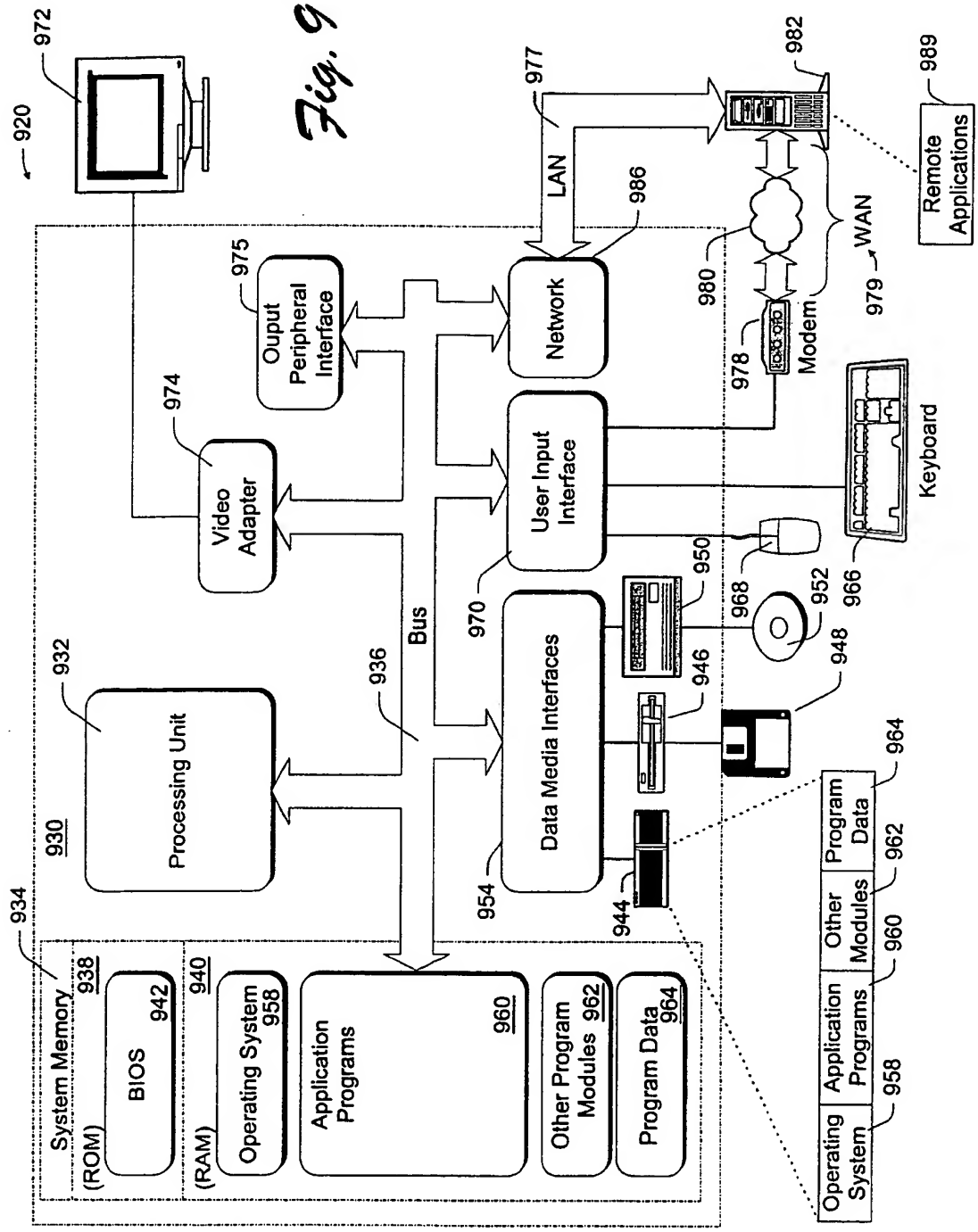

920
972
934
930
932
936
Bus
938
940
System Memory
(ROM)
BIOS
942
(RAM)
Operating System
958
Application Programs
960
Other Program Modules
962
Program Data
964
Processing Unit
Video Adapter
974
Ouput Peripheral Interface
975
Network
977
LAN
986
User Input Interface
970
Data Media Interfaces
954
944
946
948
950
952
968
966
Keyboard
982
980
978
Modem
979
WAN
989
Remote Applications
Operating System
Application Programs
Other Modules
Program Data
958
960
962
964

Fig. 9

INTERNATIONAL SEARCH REPORT

Interna' al Application No
PCT/US 00/19481

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04H1/00 G11B20/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04H G11B H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ZHAO J ET AL: "A generic digital watermarking model" COMPUTERS AND GRAPHICS, vol. 22, no. 4, July 1998 (1998-07) - August 1998 (1998-08), pages 397-403, XP002149472 UK	1,2,5-7, 9-12,14, 15, 20-22, 24,27-29
A	the whole document	3,4,16, 25,26

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
6 October 2000	30/10/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Ogor, M

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Interna al Application No
PCT/US 00/19481

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 840 513 A (NIPPON ELECTRIC CO) 6 May 1998 (1998-05-06) abstract column 2, line 10 - line 38 column 4, line 53 -column 5, line 56 column 6, line 29 - line 50 column 7, line 57 -column 8, line 12 column 9, line 36 - line 43 column 10, line 26 - line 35	1,2,5,7, 9-11,16, 20-22, 24,27, 29-32,34
A	column 10, line 46	6,12,14, 17,28

X	WO 99 11020 A (DELP EDWARD J III ;GLOGAU JORDAN J (US); LIN EUGENE TED (US); PURD) 4 March 1999 (1999-03-04) page 1, line 23 -page 2, line 17 page 3, line 3 - line 10 page 4, line 29 - line 32	1,2,6,9, 10,12, 20-22, 24,28
A	page 5, line 9 - line 20	3,4,25, 26

A	US 5 917 914 A (SHAW YIH-SUEY ET AL) 29 June 1999 (1999-06-29)	

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 00/19481

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0840513 A	06-05-1998	US 5915027 A	22-06-1999
		AU 721462 B	06-07-2000
		AU 4434097 A	07-05-1998
		CA 2219205 A	05-05-1998
		JP 10145757 A	29-05-1998
		SG 63773 A	30-03-1999
WO 9911020 A	04-03-1999	NONE	
US 5917914 A	29-06-1999	NONE	